Vehicle Start-up Instructions:

1. Navigate to Network.h file and change the variable, server\_ssid, to the corresponding vehicle in the following format. Note that each name must be unique.
   1. “veh\_X”, where X = vehicle number for example: veh\_4
2. Calibrate the compass for each vehicle
   1. Download the software MotionCal
   2. Using a spare ESP-32 board or another microcontroller, upload the calibration code from the project compass\_calibration.
   3. Plug in the magnetometer into the microcontroller using a breadboard.
   4. Open MotionCal and plug in the microcontroller into the computer. Select the port for the microcontroller. Note that you will see red dots appear on the screen when you move the magnetometer module if it is working.
   5. Move the compass module in every direction until “Gaps” reaches less than 3% but preferably less than 1%. This step may take a minute or two to achieve that gap level. Make sure to turn it upside down and do a 360 turn multiple times. It is also crucial that you do this in the position the magnetometer will be operated.
   6. Lastly, depending on the position of the magnetometer it could be off by 90 degree increments so in order to fix this go to main.cpp and change the variable named heading\_offset.
3. Upload CMPE450\_ics\_proc-Inter-com code to the ESP-32 board
4. Plug in all wire connections as described in the SDD

Setting up Leader Vehicle:

1. Using any Wifi capable device connect to the WiFi SSID “veh\_#” . Note that android devices could have trouble connecting or seeing this WiFi connection as the channel might need to be changed.
2. Open up a web browser on the same device and search “222.222.2.2” in the address bar. The controller should be displayed at this point
3. Using the dropdowns select the correct order of vehicles and press set. Note the status light should go from blinking red to green indicating the vehicles have connected to one another.
4. At this point, the joysticks will control the leader vehicle and the followers will be connected and transmitting data.